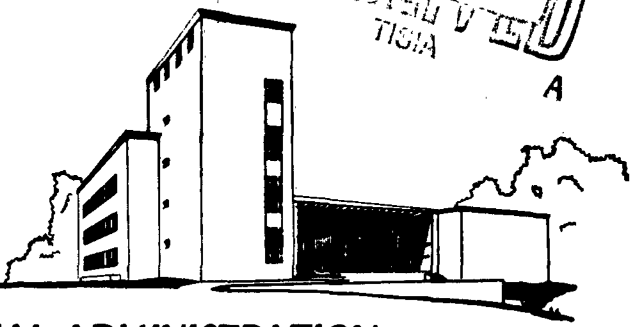
63-3-2

Carnegie Institute of Technology

Pittsburgh 13, Pennsylvania



GRADUATE SCHOOL of INDUSTRIAL ADMINISTRATION

William Larimer Mellon, Founder

ONR Research Memorandum No. 107

A HEURISTIC APPROACH TO SOLVING TRAVELLING SALESMAN PROBLEMS*

by

Robert L. Karg[1]

and

Gerald L. Thompson[2]

January 5, 1963

*The authors would like to thank Professor W. W. Cooper for valuable advice given during the preparation of this paper.

[1] National Tube Division, U. S. Steel and Duquesne University.

[2] Graduate School of Industrial Administration, Carnegie Institute of Technology.

This paper was written as part of the contract, "Planning and Control of Industrial Operations," with the Office of Naval Research and the Bureau of Ships, at the Graduate School of Industrial Administration, Carnegie Institute of Technology. Reproduction of this paper in whole or in part is permitted for any purpose of the United States Government. Contract Nonr-760(01), Project NR-047011.

1. INTRODUCTION

Let $A = \|a_{ij}\|$ be an nxn matrix of real numbers. The travelling salesman problem asks for an acyclic permutation $(i_1, i_2, \ldots, i_n)$ of the integers 1, 2, ..., n such that the sum

$$(1) \qquad a_{i_1 i_2} + a_{i_2 i_3} + \cdots + a_{i_n i_1}$$

is a minimum. If $a_{ij} = a_{ji}$ for all i and j, the problem is said to be <u>symmetric</u>; otherwise, it is <u>nonsymmetric</u>. In case the distances a_{ij} are computed between points in the Euclidean plane the problem is <u>Euclidean</u>. A Euclidean problem is, of course, symmetric.

In the nonsymmetric case the solution involves finding one out of $(n-1)!$ possible permutations, and in the symmetric case it involves finding one out of $\frac{1}{2}(n-1)!$ possible permutations. Hence, complete enumeration and evaluation of all the possible permutations provides a theoretically satisfactory solution to the problem. What is actually desired, however, is a computationally practical method of finding the optimum. Even methods that find "good" permutations would be of practical interest as approximate solutions.

The name "travelling salesman" is applicable when one interprets the sum (1) as the total distance that must be travelled by a salesman who must visit each of n cities exactly once before returning home. The acyclic requirement prevents a solution involving several disconnected loops. The travelling salesman interpretation usually results in a symmetric problem. However, if we think of a school bus that has a given number of corners at which to stop to pick up children in a city which has a substantial number of one way streets, a nonsymmetric problem results, since the distance from a to b may well be different from the distance from b to a. Still another nonsymmetric interpretation is that of a machine tool that performs a give set of jobs repetitively.

The distance from job a to job b is the setup cost, which can be different from the setup cost going from job b to job a. For instance, in a paint mixer, it is easy to go from a light to a darker color, but difficult to go in the reverse direction There are numerous other practical interpretations of this seemingly frivolous problem.

In this paper we shall discuss a method, suitable for electronic computers, that has proved capable of quickly obtaining solutions for problems having about 60 cities or less in symmetric and some nonsymmetric problems. In principle the method can be used for any size problem. Although the code does not guarantee finding the optimum tour, it can be used over and over several times and in various ways to get a probabilistic idea of how good the best answer found is relative to the set of observed answers. Also the checking procedure of Dantzig, Fulkerson, and Johnson [2] can be applied to check on the optimality of the observed result.

We call the method a heuristic one because (a) the code for it contains probabilistic elements so that its performance varies each time it is run; (b) in certain cases it can be proved that it has positive probability of producing the optimum answer, and our experience leads us to believe that it will always do so; and (c) as the result of initial calculations partial answers and sub-problems are obtained so that later calculations depend upon the results of early calculations, i.e., the process is one of "learning from experience." Nevertheless, the code is an algorithm in the sense that it terminates after a finite number of steps and has been run on an electronic computer. This application is an example of artificial intelligence, that is, the use of a computer to solve problems, the solution of which by human beings would be regarded as intelligent acts. Humans are not good at solving travelling salesman problems because of limited arithmatic abilities. Hence our code does not imitate the behavior of humans and is not heuristic in that sense.

A good summary of the history of the problem up to 1956 is presented in Flood [5]. More recently, Tucker [6] and Dantzig [4] have given integer programming formulations of the problem and some computational experience has been gained with them. The largest problem solved so far in the literature is the 42 city problem of Dantzig, Fulkerson, and Johnson [2]. Our algorithm also solved the same 42 city problem in 4.5 minutes on a Bendix G-20 computer. A much more difficult problem involving 57 cities was solved in 35 minutes. Other experience with these and smaller problems is reported on later in the paper.

In Section 2 we describe the simple idea needed for the basic step in the program. The initial code which does not have built in learning is discussed in Section 3. The final code together with its learning aspect is discussed in Section 4. In Section 5 we discuss the probabilistic methods we use for evaluating answers obtained, and in Section 6 we prove that the optimal tour has positive probability of being chosen with the algorithm.

2. THE BASIC STEP OF THE ALGORITHM

In the Euclidean travelling salesman problem it can be proved that an optimal tour will never cross itself. This follows from the Euclidean theorem that the sum of two sides of a triangle is greater than the third side. However, when cities on a map are used, the curvature of the earth, the existence of mountains having passes and tunnels, and the existence of lakes, oceans, and other natural barriers, negate the above Euclidean result. Of course, nothing like it need be true for nonsymmetric problems. In any case, if an optimal tour can be found by any means, it will have the desired properties without specifically stating restrictions needed to insure getting them.

For this reason, the only specific restriction on the permutations we construct is that they be acyclic. We have devised an inductive method for constructing acyclic permutations. The method is described in the following series of steps:

1. Choose any two cities and list them arbitrarily to form an acyclic permutation $(i_1\ \ i_2)$ of length two.

2. Assume that a permutation $(i_1\ \ i_2\ \ldots\ i_k)$ of k cities, where $2 \leq k < n$ has been constructed. Choose one of the remaining cities, call it city h. For j running from 1 to n compute the quantities

$$d_j = a_{i_j,h} + a_{h,i_{j+1}} - a_{i_j,i_{j+1}}$$

where we define i_{n+1} to be i_1 when $j = n$.

3. Let j^* be any value of j such that d_{j^*} is a minimum of the quantities computed in 2.

4. Relabel i_j as i_{j+1} for $j = j^*, \ldots, n$ and label h as i_{j^*}.

5. We thus have constructed a permutation $(i_1, \ldots\ i_{k+1})$ of k+1 cities. If $k+1 = n$ stop; otherwise replace k by k+1 and return to step 2.

Briefly, the method consists of starting with any pair of cities as a permutation of length 2, then inserting a third city in such a way as to minimize the length of the resulting tour on three cities; then inserting a fourth city in such a way as to minimize the resulting tour on four cities, etc. This is our heuristic rule for constructing acyclic permutations.

The tour resulting from this acyclic permutation may or may not be the optimum one. In fact, there are a whole set of possible tours of various lengths that can be so generated, depending upon the order in which new cities are introduced. In Section 6 we present a proof of the fact that, in the Euclidean case, there always exists an order in which to introduce the cities so that the above heuristic rule will produce the optimal tour. We have nearly always succeeded in getting what we believe to be the optimal tour even in the non-Euclidean cases and so we conjecture that the same result holds for them. However, we do not have a proof of the fact at the present time.

The five city problem of Figure 3 will help illustrate the method as well as its difficulties. Here the optimal tour is 12345, which has length 148. The only other tour constructed by our algorithm is 12453, which has length 152.

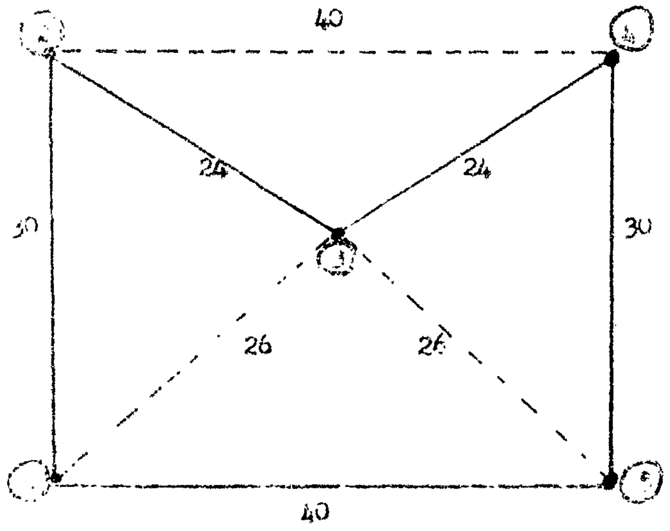


Figure 3

We ran 25 trials of the algorithm, more fully described in Section 3, using random orders for introducing new cities, and found that 15 of them resulted in the optimal tour, while the other 10 produced the suboptimal tour. Thus, for this problem, there is empirical probability of .6 of finding the optimum tour by this random procedure.

3. THE FIRST ALGORITHM CODE

The result of the algorithm of the previous section depends on the order in which the cities are introduced in step 2. So that final results were not biased we first arranged the cities in a random order list and selected cities from this list when needed for step 2. We did this a number of times, printing each tour and its distance each time. We call the first algorithm Code 1. For instance, in the 10-city problem studied by L. L. Barachet [1], whose data is given in Exhibit J and whose optimal tour is given in Exhibit B, the distribution of completion times is given in Figure 2. Note that the empirical probability of getting the optimum tour (whose length is 378 miles) is .16. Note also that the distribution is multi-modal and has gaps.

Some other experimental results are displayed in Figure 3.

No. of Cities	5	10	33	42	57
Empirical Prob. of getting optimum with Code 1.	.60	.16	.01	.0045	---

Figure 3

The probability estimates for the 5, 10, 33, and 57 were based on runs of 100 each and that of the 42 city problem on a run of 225. The shortest schedule observed in the 57 city case was 331 miles longer than what we believe to be the optimum. We did not feel that the empirical probability of getting the best answer in the 57 city case was high enough to continue computations with Code 1 until one was observed. The time to construct one tour in the 57 city case was about 30 seconds, and the time for the other problems proportionately less.

From the experience we obtained, particularly on the 33 and 42 city problems, we found that the shortest tours so produced tended to agree pretty well around the periphery of the tour where it tended to be convex, but did not agree at all well in the "center" of the problem where the optimal tour was necessarily quite non convex. Hence, we built into Code 1 the added

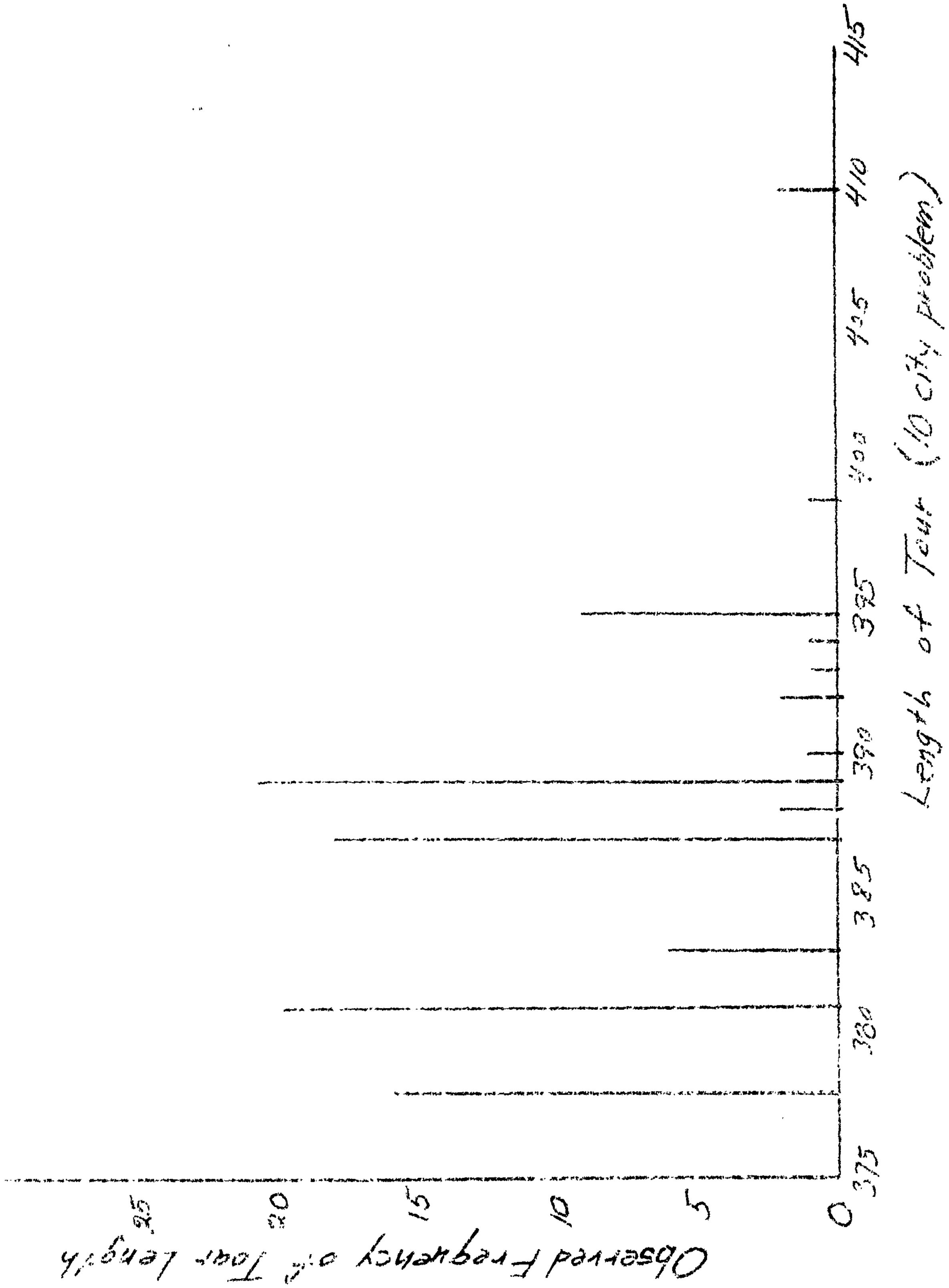

Length of Tour (10 city problem)
375
380
385
390
395
400
405
410
415
Observed Frequency of Tour Length
0
5
10
15
20
25

Figure 2

flexibility of being able to specify subproblems and have the program work on them separately.

For instance, in Figure 4 we have illustrated a case in which the eight cities at the right form a convex part of the tour that was traversed by most of the shortest tours produced in the initial runs of the problem. For this case we made a "cut" from i_1 to i_8 and solved the two sub-problems separately. In this way we "factored" the n-city problem into an 8-city and an n-6 city problem (since cities i_1 and i_8 occur in both problems). The 8-city problem, being a convex (or nearly convex) problem, is extremely easy to solve with Case 1 and usually resulted in probability of .1 or greater for the empirical frequency of observation of the shortest tour. Thus, as will be explained in Section 5, we were virtually certain of obtaining the correct answer to that sub-problem. The n-6 city problem can be considered as a new problem that is susceptible to further factoring of the same kind. The only requirement on the solutions to the series of subproblems so defined is that the common link (in Figure 4, the common link is shown dashed from i_1 to i_8) must be traversed once in the solutions to both subproblems. In the case of the non-symmetric problem, there is the further requirement that the common

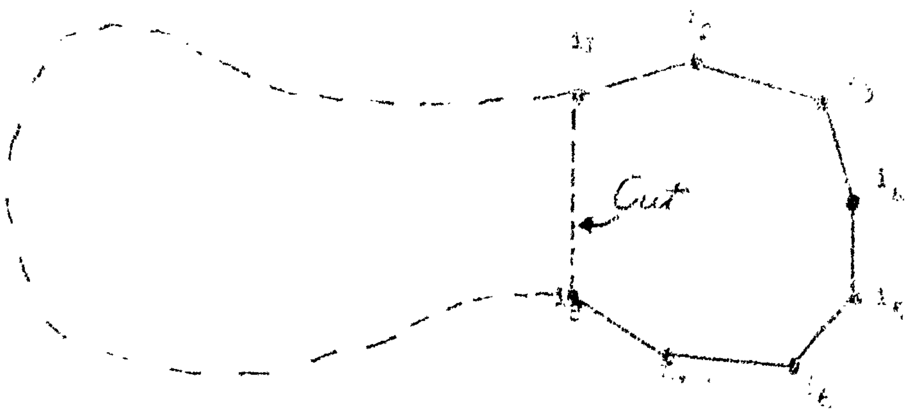


Figure 4

line be traversed once in one direction in the solution to the first subproblem, and once in the opposite direction in the solution to the complementary subproblem.

Our experience, as exhibited in Figure 5, with the 42 and 33 city problems was that we could obtain the optimal solution to each problem using all the cities. But the empirical probability of finding them in this way was quite small. We then verified probabilistically (see Section 5) that the best tour was optimal, by factoring them into subtours and showing that as subproblems these subtours had empirical probability of .1 or greater of being picked up by Code 1.

Our next idea was to have the computer do the factoring of the problem, and we describe how it did that in the next section.

4. THE LEARNING ALGORITHM--CODE 2

Our success in factoring problems after making some initial runs with Code 1, although satisfactory, was time consuming, and led us to incorporate the factoring procedure into the code. The result is Code 2 to be described next. It is a code that has learning aspects in that the results of early computations of the program define the subproblems chosen to be worked later in more detail. Of course, both correct and incorrect learning are possible. The algorithm we thus developed is one that stops after a finite number of steps and prints out a tour that may or may not be optimal. Our experience shows that it has positive probability of finding the optimal tour which depends upon the size of the problem being worked and on the number of random tries the machine is permitted to make (i.e., the experience it is allowed to have) before being required to define subproblems to be worked on. To describe just one numerical result with Code 2 we found that in the 42 city problem the algorithm found the correct answer after making 5 cuts, i.e., it defined 6 subproblems. The cuts can be observed in Exhibit D. Other results will be discussed later.

We describe in more detail the algorithm for Code 2 which includes the algorithms previously described in Sections 2 and 3.

0. Read initial data.

1. Choose two link cities, x and y, at random from among the list of possible cities.

2. Eliminate the link cities from the list. Then put the remaining cities in random order

3. Using the link cities, x and y, as the initial acyclic permutation on 2 cities, use the algorithm of Section 2 to construct a permutation on n cities and compute the length of the tour.

4. Go back to 2 and construct a new permutation. Compare each time with the previous shortest tour found, saving the best one observed in k trials, where the value of k is read in as data, or else k is chosen to be a multiple of n. For instance, k = 50 or 75 or k = 2n are typical values we used.

5. Print the best tour found in k loops.

6. Find a city on the diameter of the best tour as follows: Choose any city, a, at random; find the city, b, on the tour that is farthest from a; then find the city, c, on the tour that is farthest from b. Select c as the diameter city. (This process could be repeated more often, but would cycle. The net effect is to get a city on the tour that is "far away" from the "center" of the problem.)

7. Define a convex subproblem. To describe how we do this, let us assume that diameter city c is also i_{10} the 10th city on the best tour found in 4.

We now compute the distance d from i_{10} to i_9. Next we compute the distance d' from i_9 to i_{11} and see whether it is greater than or equal to d. In any case, we replace d by d' and go on. Next we compute the distance d'' from i_{11} to i_8 and make the same comparison, replacing d by this distance. We continue this process as long as d continues to increase. Once d starts to decrease, we continue the process as long as d continues to decrease, and terminate it as soon as d starts to increase again. Thus, if the optimal tour consists of points on a circle we would include the whole problem as a subproblem. But if the optimal tour were in the shape of an hour glass, as in Figure [illegible], we would cut off one lobe of [illegible] etc. Illustrations of subproblems defined by the program are shown in Exhibits C, D, and E. The reader will note in these figures that the resulting subproblems are not always convex in the strict mathematical sense. However, the resulting subproblems are sufficiently simple for our algorithm to solve easily, which is all that is desired.

8. Once the convex subproblem is defined, we print out the partial answer. If the convex subproblem includes all the cities of the original problem we stop the procedure. If the subproblem does not include all the cities on the list, we set up a new problem that consists of all the cities not included in the convex problem printed out, determine the link cities x and y (being careful about the order of these cities in the nonsymmetric case) and go back to 2. Since a finite number of cities are removed each time the cities of a subproblem are deleted from the list of remaining cities, this process will stop after a finite number of steps.

In another version of the program we employed "double-cutting," that is, removing a subproblem from both ends of the diameter of the partial tour chosen in 6. This process worked reasonably well but necessitated a longer learning loop in 4 to be certain that both ends of the periphery of the tour were correct.

There was a net saving of computational time, however.

It should be obvious to the reader that the program we have outlined was devised by looking at geometrical, hence symmetric and euclidean, problems. Nevertheless, the program is entirely arithmetical in nature, and the limited experience we have had with perhaps 5 nonsymmetric problems indicated that it works for these problems as well. However, the latter conclusion is highly tentative and needs a good deal of further study and experimentation before it can be firmly asserted.

Our experience with the various sample problems is shown in Figure 5. The actual cuts made by the machine are shown dashed in Exhibits A-F. In Figure 5 we have indicated the experience both with and without the double-cutting feature.

Of these problems the 5 and 10 city problems were completely trivial to solve. The 33 city problem was interesting in that the program (with single cutting), had mistakes in its initial guesses that were not corrected until the last cut was made. In the 42 city problem (with single-cutting), the program actually obtained the correct answer after the first cut, so that doubtless a shorter learning loop would also have produced the correct answer. Note that double-cutting almost halved the time needed to solve this problem. The answers shown to these four problems have been proved to be the optimal answers by various people [2, 3].

By far the most difficult problem is the 57 city problem, the data for which was obtained from the Rand-McNally 1962 road atlas of the United States. On some initial runs of the problem we obtained a tour that had length 12,966 miles shown in Exhibit F. The best answer that we found is that shown in Exhibit E which has length 12,985 miles. Although these two tours agree pretty well around the periphery, they are quite different in the middlewest. An answer to this problem that is 30 miles shorter has been obtained by Gordon and S

No. of Cities	5	10	33	42	57
Length of Learning Loop, Code 2 (single cutting)	10	20	50	75	150
Number of cuts	0	0	3	5	6**
Time	0:02*	0:11	2:29	7:21	33:31
Length of Learning Loop, Code 2 (double cutting)	-	-		2n	3n
Number of cuts	-	-		4**	6**
Time	-			4:27	16:56***

*Estimated Time

**Additional cuts were made, but not needed

***The answer was the same as in Exhibit E except for the mistake in the Northeast, discussed in Section 5.

Figure 5

Reiter (private communication). In spite of repeated attempts our program has never produced their solution, the probable reasons for this are discussed in Section 6. It should be noted that their procedure requires several days of computation to obtain such an answer, so that in terms of cost of computation the answer in Exhibit E is still probably preferable.

In addition to the experience already discussed we have made several runs on larger problems with randomly generated data of dimensions as high as 90 x 90. Although the program works for such problems we have no way of comparing how good the answers are with any other standard so that we shall not report on such experience here.

5. CHECKING OF ANSWERS

One method for checking the optimality of proposed answers to travelling salesman problems has been given by Dantzig, Fulkerson, and Johnson [2]. That method is, of course, applicable to the answers which our program gives.

We propose here a probabilistic method for checking on the accuracy of the result. Our model is the simple binomial trials model in which there are two events success and failure. In Figure 6 we have listed the probabilities of observing successes in n trials where p is the probability of a success each time.

p \ n	10	25	50	75	100	200
.100	.651322	.928210	.994846	.999630	.999973	1.000000
.050	.401263	.722610	.923055	.978656	.994079	.999960
.020	.182927	.396535	.635830	.780256	.867380	.982412
.010	.095618	.222179	.394994	.529413	.633968	.866020

Figure 6

From the table it can be seen that events of probability .1 or greater are essentially certain of being observed in 50 trials and events of probability .05 or more are essentially certain of being observed in 100 trials. Our own philosophy is to delete the word "essentially" and regard these events as certain. This is an approximation, and our solutions are approximate in this probabilistic sense. Nevertheless, there are several ways of improving on the confidence that one feels in the answer so obtained. We list some of these methods next.

(a) Run Code 2 several times to get different ways of cutting the problem up and different answers. Take the best answer of these.

(b) Cut the best answer manually into subproblems that are different from the ones that Code 2 used. Apply Code 1 to the subproblems.

(c) Take adjacent subproblems, i.e., ones that have a common link and work pairs of them using Code 1 to see if the subproblems define the same answer as that given by Code 1.

(d) From Figure 6 it is possible to estimate the probability of the various subproblems being worked correctly. By taking the product of these probabilities it is possible to estimate the probability of having found the complete tour at random using Code 1, and thus get an idea of how much has been "learned" from the cutting and factoring procedures of Code 2. The probability thus obtained is a good upper estimate of the probability of finding the optimum tour, and can be used to estimate the cost of getting a better tour than the one found so far.

(e) Of course, it is obvious that using long learning loops in Code 2 and repeating it a number of times will improve the reliability estimate that one can put in the final best answer found by the algorithm.

We do not claim that our program is infallible, but rather that it gives good answers in a computationally feasible amount of computer time. For instance, a better tour (30 miles shorter) is known for the 57 city problem than any that our programs found. An explanation of the failure of our program to find the better one may be found by examining the nine city subproblem consisting of the cities 34, 42, 7, 6, 36, 39, 19, 3, and 56 in the northeastern part of the United States (see Exhibits E and F). The optimum tour and another tour that is 147 miles longer are the two most probable tours chosen by our heuristic. The relative probabilities of choosing these tours, obtained by running Code 1 500 times on this subproblem are shown in Figure 7.

Tour	Observed Probability
3-19-39-36-6-7-42-34-56	.192
3-56-34-42-7-19-6-36-39	405

Figure 7

Thus the non-optimal tour is twice as probable as the optimal tour, and actually will be chosen about 40 percent of the time. It happened in many of the runs that we made that the best tours found would contain the non-optimal tour in the north-east, and this error being on the periphery of the best tour observed caused difficulties, particularly when double-cutting was used. This tendency of our program to choose highly probable schedules, relative to the heuristic used, constitutes a weakness of the method. It is undoubtedly for this reason that we were unable to observe the Gordan-Reiter tour which was 30 miles shorter, i.e., the tours in Exhibits E and F were probably chosen much more often by the algorithm, and the shorter tour is extremely unlikely to be observed in the relatively short amounts of computer time we used, as compared to the times employed by Gordan and Reiter.

6. PROOF THAT THE OPTIMAL TOUR CAN BE CHOSEN IN THE EUCLIDEAN CASE

In the Euclidean case it is well-known that an optimal tour can never cross itself. This follows from the triangle inequality, for if a tour is used that crosses itself then it is easy to see how to pull it away at the crossing point and shorten the tour.

Now consider an optimal tour T_n in an n-city travelling salesman problem in the euclidean plane. We first show that the interior of the tour T_n can be triangulated by means of line segments lying completely in the interior of T_n. For instance, in Figure 8 we illustrate an eight city optimal tour that is so triangulated.

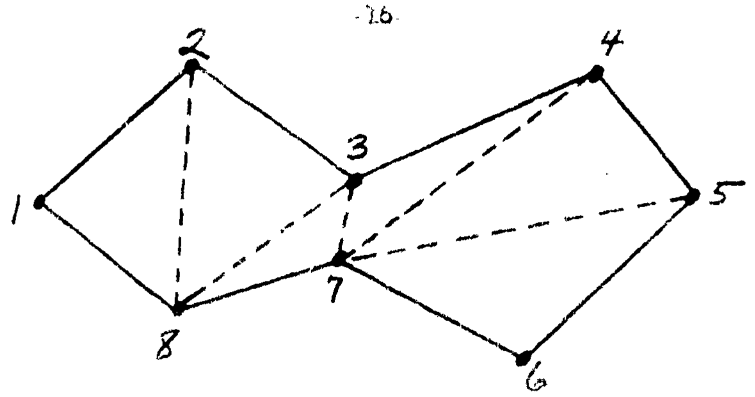


Figure 8

The line segments in the interior of the tour are shown dashed. Note that the inside of the tour has been decomposed into triangles.

To prove that this can always be done, observe that it is vacuously true for T_3, and the only two possible cases for T_4 are shown in Figure 9. Now assume that all tours T_{n-1} on n-1 cities can be so triangulated. Consider an optimal

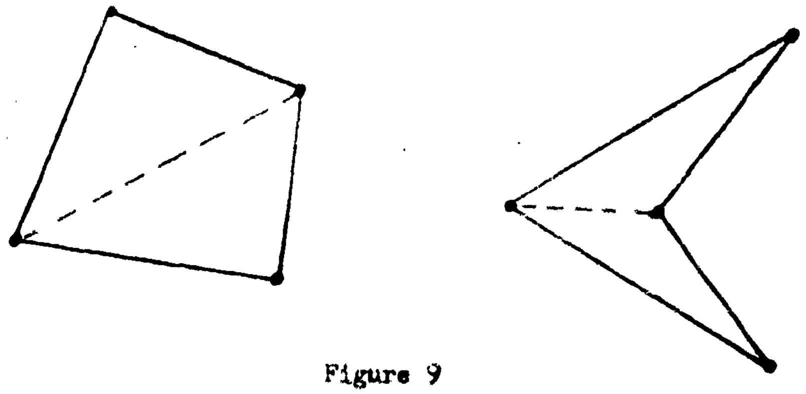


Figure 9

tour T_n for n cities. Choose a vertex, call it i_1, at which the angle formed by the two edges of the tour that meet at i_1 and which points into the interior of the tour, is less than 180°. Consider the cities i_2 and i_{n-1} which are adjacent to i_1 in the optimal tour, as shown in Figure 10. We have two cases: (a) the line segment joining i_2 and i_{n-1} lies entirely inside the tour; or (b) the line segment does not lie entirely within the tour. In case (a) we include the line segment from i_2 to i_{n-1} as part of the triangulation, and now the tour that bypasses i_1 by making use of this line segment consists of

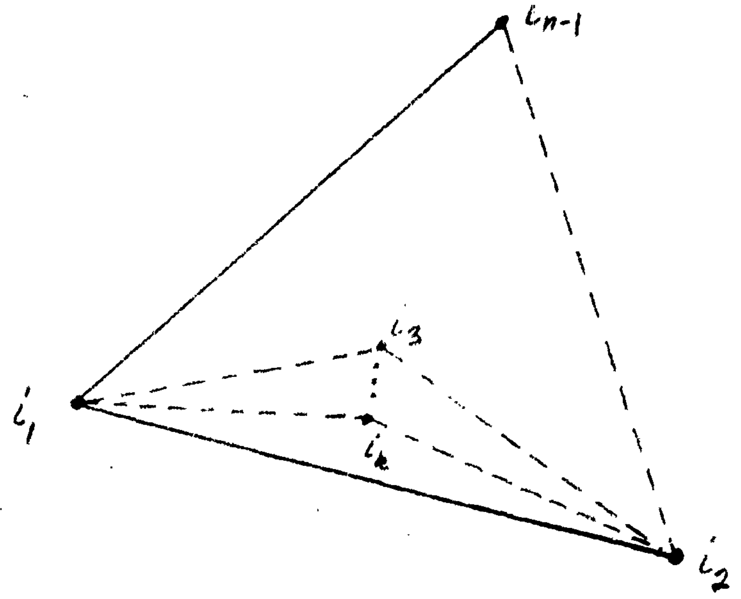


Figure 10

n-1 cities. This tour must be optimal for that problem or else the original tour on n cities could be shortened. Hence by the induction assumption the remaining tour can be triangulated, giving a triangulation of T_n. In case (b), there must be a city, call it i_3, in the interior of the triangle formed by i_1, i_2, and i_{n-1}. We now consider the line segment from i_1 to i_3. Either it lies completely inside T_n or else there is another city i_4 in side the triangle formed by i_1, i_2, and i_3. Continuing in this way we eventually find (since there are only a finite number of cities) a line segment from i_1 to some other city on the tour, say i_k, and which lies entirely inside the optimal tour T_n. This line segment divides T_n into two sub-tours each involving fewer than n cities. Hence by the induction assumption these two sub-tours can be triangulated, and these give a triangulation of T_n.

By the same kind of inductive argument, it can be shown that n-3 interior segments will be needed to perform the triangulation. It can also be shown that there is at least one vertex that has no interior segment connected to it, such as vertex 1 in Figure 8.

To demonstrate that there is a random order of the cities that will make the heuristic produce the optimal tour, we consider a triangulation of the optimal tour T_n. Remember the cities so that city 1 is a vertex having no interior segment of the triangulation connected to it (as in Figure 8), and assume that the other cities are numbered in order around the tour Then, necessarily because we have a triangulation, cities 2 and n will be connected by an interior segment (2 and 8 in Figure 8). Hence we make cities 1 and 8 be the first two cities considered by the algorithm, and city 2 the next one on the list. The segment from 2 to n will be the base of one of the triangles of the triangulation; let the next city on the list be the city at the peak of that triangle, etc. For instance, in Figure 8 we let cities 1 and 8 be the initial cities, and introduce the other cities in the order 2, 3, 7, 4, 5, and 6. The algorithm given in Section 2 will then produce the optimal tour 1, 2, 3, 4, 5, 6, 7, 8, as the reader can easily check. Thus for every different triangulation of the optimal tour we get a different initial list that will produce the optimal answer. There are other initial lists which will also produce the optimal tour, for instance the list 1, 8, 5, 4, 6, 3, and 7 in Figure 8.

At present, we do not have a proof that for non-euclidean problems there is an initial list which will produce the optimal tour. Nevertheless, our experience leads us to conjecture that such is possible.

Bibliography

[1] Barachet, L. L., "Graphic Solution of the Travelling Salesman Problem," Operations Research 5 (1957) 841-845.

[2] Dantzig, G. B., R. Fulkerson, and S. M. Johnson, "Solution of a Large-scale Travelling Salesman Problem," Operations Research 2 (1954) 393-410.

[3] Dantzig, G. B., D. R. Fulkerson, and S. M. Johnson, "On a Linear-Programming, Combinatorial Approach to the Travelling Salesman Problem," Operations Research 7 (1959) 58-66.

[4] Dantzig, G. B., "On the Significance of Solving Linear Programming Problems With Some Integer Variables," Econometrica 28 (1960) 30-44.

[5] Flood, M. M., "The Travelling Salesman Problem," Operations Research 4 (1956) 61-75.

[6] Tucker, A. W., "An Integer Program for a Multiple-Trip Variant of the Travelling Salesman Problem," private notes.

Exhibit A

5 CITY OPTIMAL TOUR

Distance 148

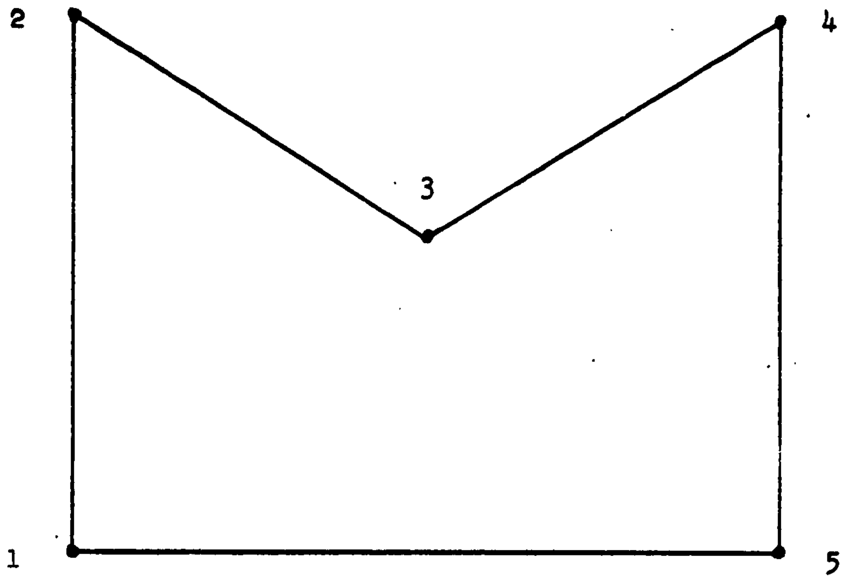


Exhibit B

10 CITY OPTIMAL TOUR

Distance 378

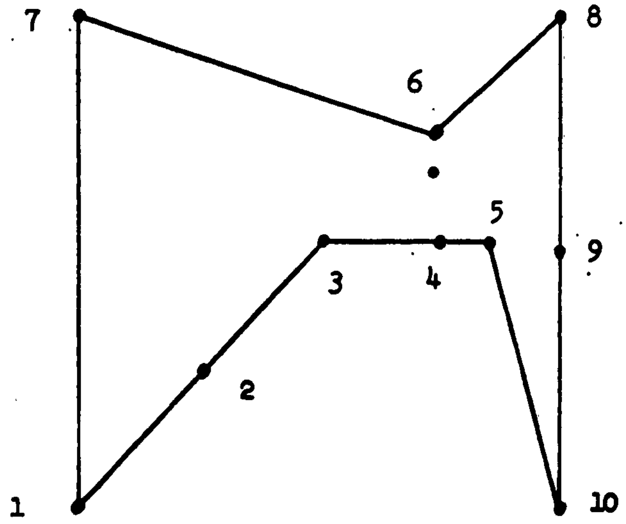


Exhibit C

BEST 33 CITY TOUR FOUND

(P&G Optimum)

Distance 10,861 miles

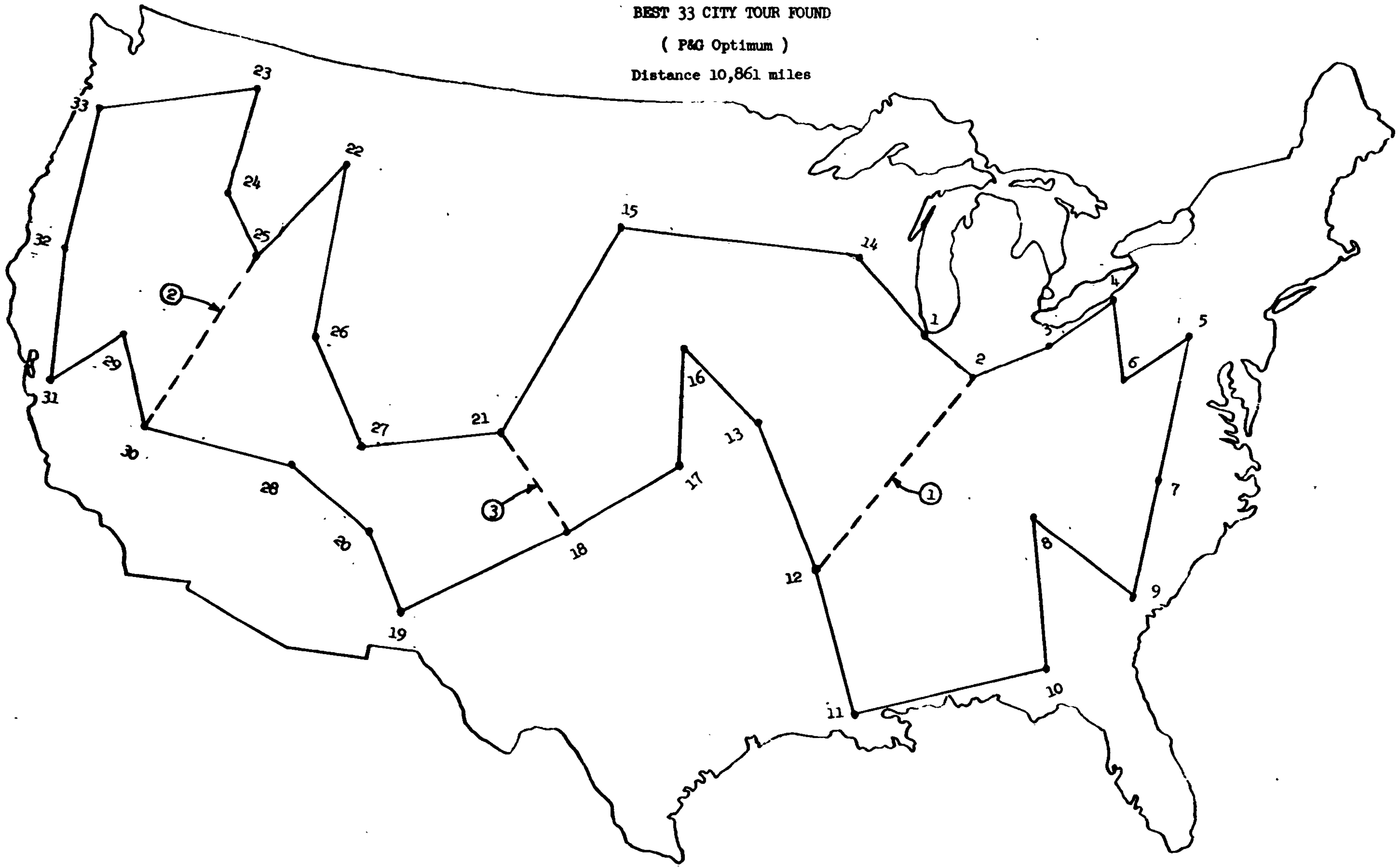


Exhibit D

BEST 42 CITY TOUR FOUND

(Optimum)

Distance 699

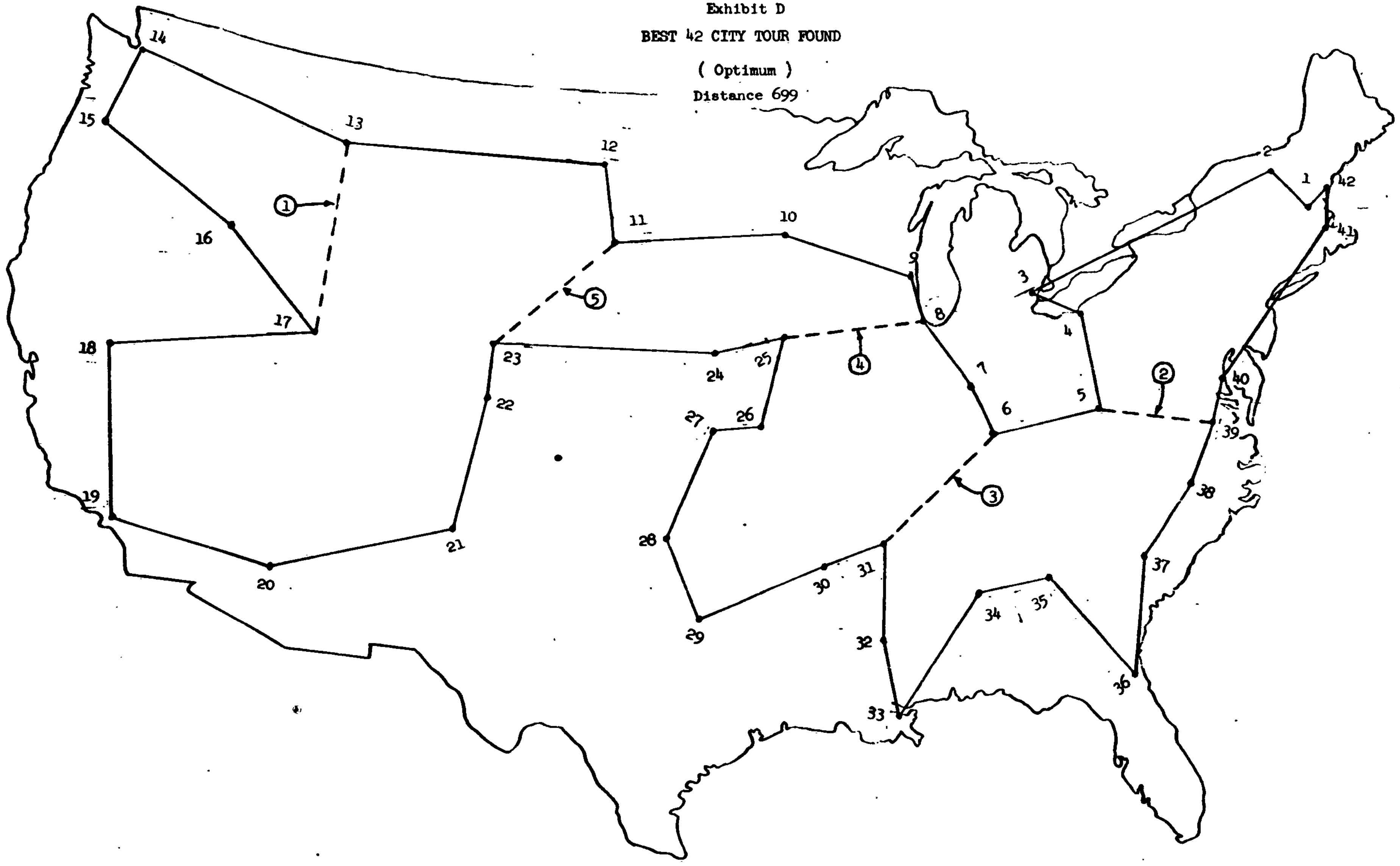


Exhibit E

BEST 57 CITY TOUR FOUND

Distance 12,955 miles

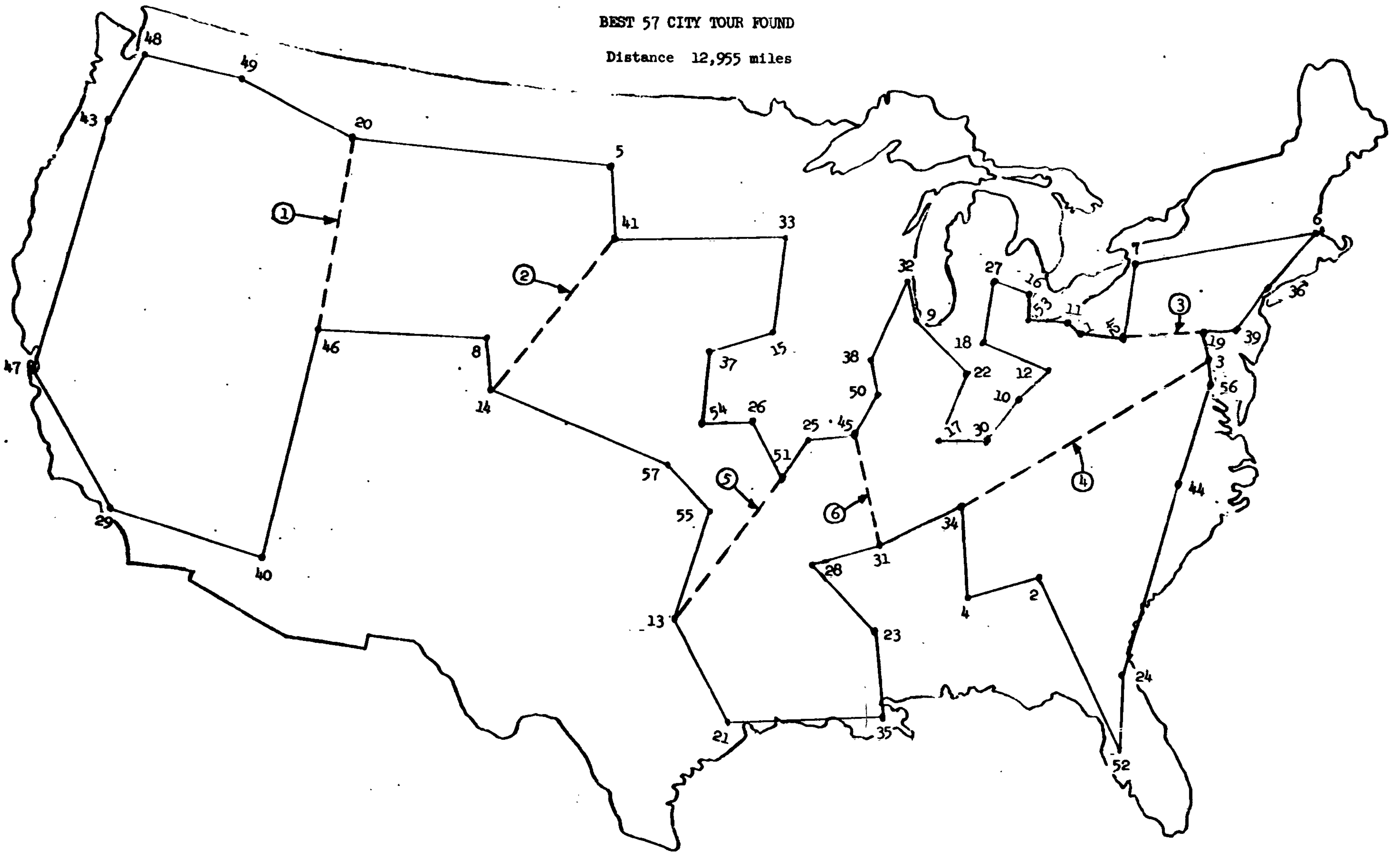


Exhibit F

NEXT BEST 57 CITY TOUR FOUND

Distance 12,956 miles

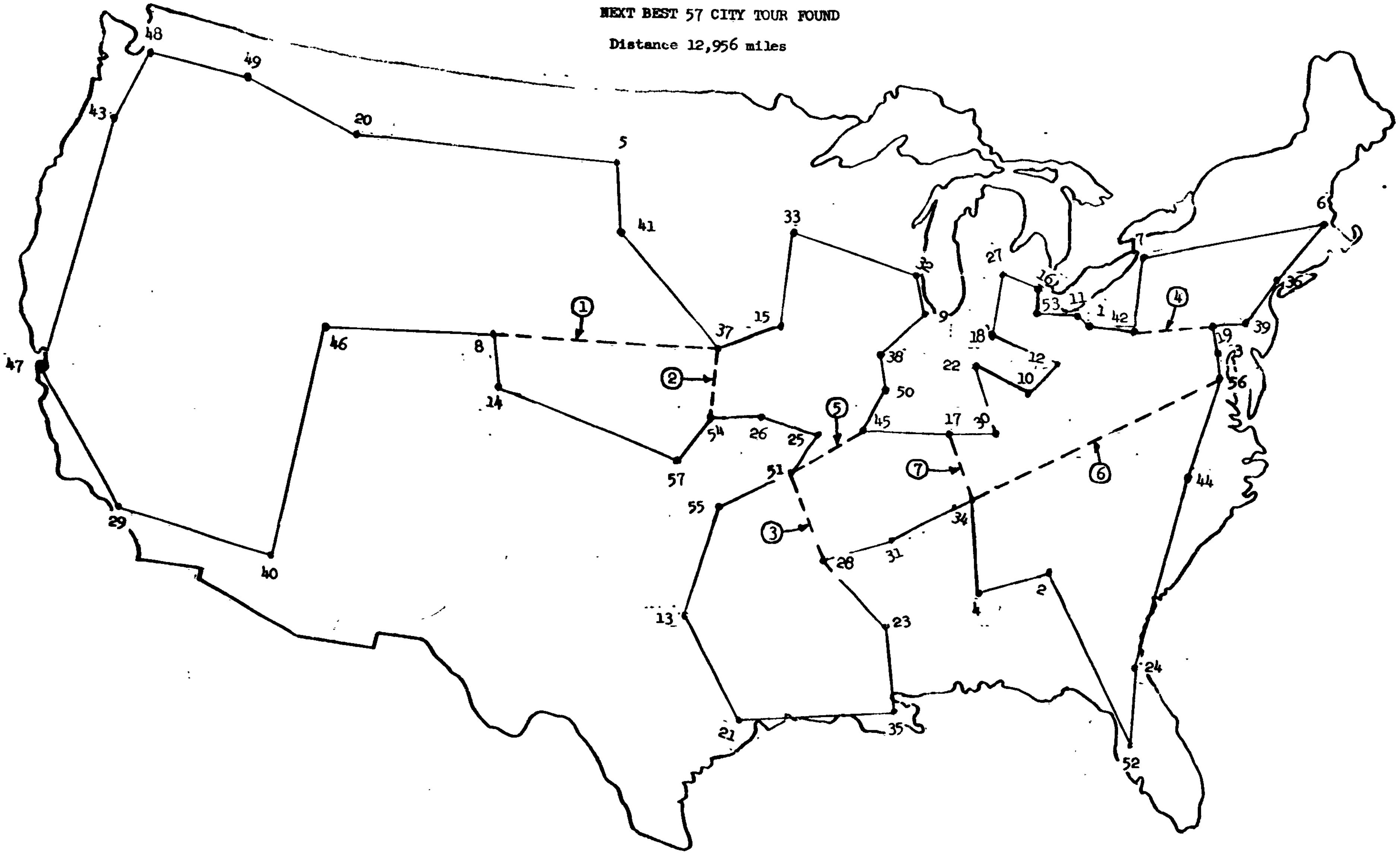


Exhibit G
List
<u>33 City Problem</u>

1. Chicago, Ill.
2. Indianapolis, Ind.
3. Marion, Ohio
4. Erie, Penna.
5. Carlisle, Penna.
6. Wana, West Virginia
7. Wilkesboro, N. C.
8. Chattanooga, Tenn.
9. Barnwell, S. Car.
10. Bainbridge, Ga.
11. Baton Rouge, La.
12. Little Rock, Ark.
13. Kansas City, Mo.
14. La Crosse, Wis.
15. Blunt, S. Dak.
16. Lincoln, Neb.
17. Wichita, Kan.
18. Amarillo, Tex.
19. Truth or Consequences, N. Mex.
20. Manuelito, N. Mex.
21. Colorado Springs, Colo.
22. Butte, Mont.
23. Lewiston, Ida.
24. Boise, Idaho
25. Twin Falls, Ida.
26. Salt Lake City, Utah
27. Mexican Hat, Utah
28. Marble Canyon, Ariz.
29. Reno, Nev.
30. Lone Pine, Calif.
31. Gustine, Calif.
32. Redding, Calif.
33. Portland, Ore.

Exhibit H
List
<u>42 City Problem</u>

1.	Manchester, N. H.	22.	Denver, Colo.
2.	Montepelier, Vt.	23.	Cheyenne, Wyo.
3.	Detroit, Mich.	24.	Omaha, Neb.
4.	Cleveland, Ohio	25.	Des Moines, Iowa
5.	Charleston, W. Va.	26.	Kansas City, Mo.
6.	Louisville, Ky.	27.	Topeka, Kans.
7.	Indianapolis, Ind.	28.	Oklahoma City, Okla.
8.	Chicago, Ill.	29.	Dallas, Tex.
9.	Milwaukee, Wis.	30.	Little Rock, Ark.
10.	Minneapolis, Minn.	31.	Memphis, Tenn.
11.	Pierre, S. D.	32.	Jackson, Miss.
12.	Bismark, N. D.	33.	New Orleans, La.
13.	Helena, Mont.	34.	Birmingham, Ala.
14.	Seattle, Wash.	35.	Atlanta, Ga.
15.	Portland, Ore.	36.	Jacksonville, Fla.
16.	Boise, Idaho	37.	Columbia, S. C.
17.	Salt Lake City, Utah	38.	Raleigh, N. C.
18.	Carson City, Nev.	39.	Richmond, Va.
19.	Los Angeles, Calif.	40.	Washington, D. C.
20.	Phoenix, Ariz.	41.	Boston, Mass.
21.	Santa Fe, N. Mex.	42.	Portland, Me.

Exhibit I
List
<u>57 City Problem</u>

1. Akron, Ohio
2. Atlanta, Ga.
3. Baltimore, Md.
4. Birmingham, Ala.
5. Bismarck, N. Dak.
6. Boston, Mass.
7. Buffalo, N. Y.
8. Cheyenne, Wyo.
9. Chicago, Ill.
10. Cincinnati, Ohio
11. Cleveland, Ohio
12. Columbus, Ohio
13. Dallas, Tex.
14. Denver, Colo.
15. Des Moines, Iowa
16. Detroit, Mich.
17. Evansville, Ind.
18. Ft. Wayne, Ind.
19. Harrisburg, Pa.
20. Helena, Mont.
21. Houston, Tex.
22. Indianapolis, Ind.
23. Jackson, Miss.
24. Jacksonville, Fla.
25. Jefferson City, Mo.
26. Kansas City, Mo.
27. Lansing, Mich.
28. Little Rock, Ark.
29. Los Angeles, Calif.
30. Louisville, Ky.
31. Memphis, Tenn.
32. Milwaukee, Wis.
33. Mpls, St. Paul, Minn.
34. Nashville, Tenn.
35. New Orleans, La.
36. New York, N. Y.
37. Omaha, Nebr.
38. Peoria, Ill.
39. Philadelphia, Pa.
40. Phoenix, Ariz.
41. Pierre, S. Dak.
42. Pittsburgh, Pa.
43. Portland, Ore.
44. Raleigh, N. C.
45. St. Louis, Mo.
46. Salt Lake City, Utah
47. San Francisco, Calif.
48. Seattle, Wash.
49. Spokane, Wash.
50. Springfield, Ill.
51. Springfield, Mo.
52. Tampa, Fla.
53. Toledo, Ohio
54. Topeka, Kans.
55. Tulsa, Okla.
56. Washington, D. C.
57. Wichita, Kans.

Exhibit J

Data for the Five City Problem

Source: Hypothetical Example

$$\begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{array} \begin{bmatrix} 0 & & & & \\ 30 & 0 & & & \\ 26 & 24 & 0 & & \\ 50 & 40 & 24 & 0 & \\ 40 & 50 & 26 & 30 & 0 \end{bmatrix}$$

Exhibit K

Data for the Ten City Problem

Source: L. L. Barachet, "Graphic Solution of the Travelling Salesman Problem," O. R. 5(1957) 841-5

$$\begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \\ 8 \\ 9 \\ 10 \end{array} \begin{bmatrix} 0 & & & & & & & & & \\ 28 & 0 & & & & & & & & \\ 57 & 28 & 0 & & & & & & & \\ 72 & 45 & 20 & 0 & & & & & & \\ 81 & 54 & 30 & 10 & 0 & & & & & \\ 85 & 57 & 28 & 20 & 22 & 0 & & & & \\ 80 & 63 & 57 & 72 & 81 & 63 & 0 & & & \\ 113 & 85 & 57 & 45 & 41 & 28 & 80 & 0 & & \\ 89 & 63 & 40 & 20 & 10 & 28 & 89 & 40 & 0 & \\ 80 & 63 & 57 & 45 & 41 & 63 & 113 & 80 & 40 & 0 \end{bmatrix}$$

EXHIBIT L

Data for the 33 City Problem

Source: Rand-McNally Road Atlas, 38th Edition, Rand-McNally Company: 1962

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33
1	0																																
2	184	0																															
3	292	195	0																														
4	449	310	215	0																													
5	670	540	380	288	0																												
6	516	357	232	200	211	0																											
7	598	514	434	566	436	381	0																										
8	618	434	493	787	814	642	295	0																									
9	881	697	719	790	632	697	224	320	0																								
10	909	964	955	1020	974	952	541	341	318	0																							
11	978	892	1031	1246	1352	1180	843	538	747	441	0																						
12	654	597	803	1018	1154	1104	766	461	749	634	380	0																					
13	504	503	722	937	1043	806	986	722	1042	954	784	404	0																				
14	276	460	568	725	946	817	874	894	1214	1185	1218	660	452	0																			
15	780	964	1072	1229	1450	1321	1378	1326	1646	1672	1410	1030	626	476	0																		
16	529	644	789	1004	1184	1001	1214	950	1270	1213	1043	632	219	436	419	0																	
17	805	698	917	1132	1238	1055	1113	842	1162	1027	779	473	195	637	634	256	0																
18	1181	1007	1226	1441	1547	1364	1375	1080	1134	1138	783	611	563	1046	759	624	368	0															
19	1548	1444	1630	1845	1984	1801	1726	1431	1685	1477	1134	1033	906	1389	1094	967	711	404	0														
20	1547	1454	1668	1883	1994	1811	1879	1584	1776	1632	1267	1053	944	1427	1196	1005	749	442	251	0													
21	1239	1167	1353	1568	1707	1524	1584	1313	1633	1498	1151	979	614	988	600	525	471	368	512	507	0												
22	1538	1733	1830	2045	2208	2090	2136	2078	2398	2332	2110	1782	1378	1300	760	1229	1382	1319	1163	930	910	0											
23	1999	2158	2291	2448	2669	2515	2597	2500	2820	2675	2336	2164	1707	1860	1375	1582	1658	1545	1389	1156	1237	436	0										
24	1716	1875	2008	2165	2386	2232	2488	2217	2537	2392	2053	1881	1422	1577	1106	1244	1375	1262	1106	873	954	483	283	0									
25	1580	1738	1872	2029	2250	2095	2352	2081	2401	2256	1917	1745	1286	1473	988	1147	1239	1126	970	737	818	379	419	136	0								
26	1425	1569	1717	1874	2109	1926	2115	1844	2164	2019	1680	1508	1118	1335	862	913	1002	889	733	500	581	430	656	373	237	0							
27	1560	1549	1852	2009	2089	1906	2063	1792	2112	1967	1456	1274	1032	1540	1068	1007	944	665	521	282	491	768	994	711	575	358	0						
28	1918	1744	1963	2178	2284	2101	2174	1879	2071	1892	1562	1348	1239	1722	1258	1300	1044	737	526	295	802	816	1022	739	603	386	545	0					
29	2065	2102	2357	2514	2642	2459	2626	2355	2675	2530	2191	2019	1673	1905	1432	1570	1507	1320	1109	878	1124	842	715	432	465	533	849	739	0				
30	2284	2131	2326	2441	2671	2488	2418	2123	2437	2259	1929	1715	1606	1924	1451	1589	1411	1104	893	662	1143	1004	981	698	599	589	768	523	266	0			
31	2340	2348	2543	2658	2888	2705	2869	2598	2918	2773	2434	2262	1916	2148	1675	1813	1750	1468	1102	871	1367	1085	958	675	1033	778	1092	982	243	349	0		
32	2247	2327	2539	2696	2867	2684	2851	2580	2900	2755	2416	2244	1898	2130	1657	1795	1732	1545	1334	1103	1349	1014	814	531	1015	760	1047	964	225	497	266	0	
33	2163	2322	2455	2612	2833	2679	2761	2664	2984	2839	2500	2328	1871	2024	1539	1746	1822	1709	1553	1320	1391	693	346	447	583	820	1158	1355	581	847	710	444	0

EXHIBIT M

Data for the 42 City Problem

ROAD DISTANCES BETWEEN CITIES IN ADJUSTED UNITS

The figures in the table are mileages between the two specified numbered cities, less 11, divided by 17, and rounded to the nearest integer.

Source: G. Dantzig, R. Fulkerson, and S. Johnson
"Solution Of A Large-Scale Traveling-Salesman Problem"
Journal of the Operations Research Society of America
November, 1954
Volume 2, Number 4

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41
2	8																																								
3	39	45																																							
4	37	47	9																																						
5	50	49	21	15																																					
6	61	62	21	20	17																																				
7	58	60	16	17	18	6																																			
8	59	60	15	20	26	17	10																																		
9	62	66	20	25	31	22	15	5																																	
10	81	81	40	44	50	41	35	24	20																																
11	103	107	62	67	72	63	57	46	41	23																															
12	108	117	66	71	77	68	61	51	46	26	11																														
13	145	149	104	108	114	106	99	88	84	63	49	40																													
14	181	185	140	144	150	142	135	124	120	99	85	76	35																												
15	187	191	146	150	156	142	137	130	125	105	90	81	41	10																											
16	161	170	120	124	130	115	110	104	105	90	72	64	34	31	27																										
17	142	146	101	104	111	97	91	85	86	75	51	59	29	53	48	21																									
18	174	178	133	138	143	129	123	117	118	107	83	84	54	46	35	26	31																								
19	185	186	142	143	140	130	126	124	128	118	93	101	72	69	58	58	43	26																							
20	164	165	120	123	124	106	106	105	110	104	86	97	71	93	82	62	42	45	22																						
21	137	139	94	96	94	80	78	77	84	77	56	64	65	90	87	58	36	68	50	30																					
22	117	122	77	80	83	68	62	60	61	50	34	42	49	82	77	60	30	62	70	49	21																				
23	114	118	73	78	84	69	63	57	59	48	28	36	43	77	72	45	27	59	69	55	27	5																			
24	85	89	44	48	53	41	34	28	29	22	23	35	69	105	102	74	56	88	99	81	54	32	29																		
25	77	80	36	40	46	34	27	19	21	14	29	40	77	114	111	84	64	96	107	87	60	40	37	8																	
26	87	89	44	46	46	30	28	29	32	27	36	47	78	116	112	84	66	98	95	75	47	36	35	12	11																
27	91	93	48	50	48	34	32	33	36	30	34	45	77	115	110	83	63	97	91	72	44	32	36	9	15	3															
28	105	106	62	63	64	47	46	49	54	48	46	59	85	113	115	88	66	98	79	59	31	36	42	28	33	21	20														
29	111	113	69	71	66	51	53	56	61	57	59	71	96	130	126	98	75	98	85	62	38	47	53	39	42	29	30	12													
30	91	92	50	51	46	30	34	38	43	49	60	71	103	141	136	109	90	115	99	81	53	61	62	36	34	24	28	20	20												
31	83	85	42	43	38	22	26	32	36	51	63	75	106	142	140	112	93	126	108	88	60	64	66	39	36	27	31	28	28	8											
32	89	91	55	55	50	34	39	44	49	63	76	87	120	155	150	123	100	123	109	86	62	71	78	52	49	39	44	35	24	15	12										
33	95	97	64	63	56	42	49	56	60	75	86	97	126	160	155	128	104	128	113	90	67	76	82	62	59	49	53	40	29	25	23	11									
34	74	81	44	43	35	23	30	39	44	62	78	89	121	159	155	127	108	136	124	101	75	79	83	54	50	42	46	43	39	23	14	14	21								
35	67	69	42	41	31	25	32	41	46	64	83	90	130	164	160	133	114	146	134	111	85	84	86	59	52	47	51	53	49	32	24	24	30	9							
36	74	76	61	60	42	44	51	60	66	83	102	110	147	185	179	155	133	159	146	122	98	105	107	79	71	66	70	70	60	48	40	36	33	25	18						
37	57	59	46	41	25	30	36	47	52	71	93	98	136	172	172	148	126	158	147	124	121	97	99	71	65	59	63	67	62	46	38	37	43	23	13	17					
38	45	46	41	34	20	34	38	48	53	73	96	99	137	176	178	151	131	163	159	135	108	102	103	73	67	64	69	75	72	54	46	49	54	34	24	29	12				
39	35	37	35	26	18	34	36	46	51	70	93	97	134	171	176	151	129	161	163	139	118	102	101	71	65	65	70	84	78	58	50	56	62	41	32	38	21	9			
40	29	33	30	21	18	35	33	40	45	65	87	91	117	166	171	144	125	157	156	139	113	95	97	67	60	62	67	79	82	62	53	59	66	45	38	45	27	15	6		
41	3	11	41	37	47	57	55	58	63	83	105	109	147	186	188	164	144	176	182	161	134	119	116	86	78	84	88	101	108	83	80	86	92	71	64	71	54	41	32	25	
42	5	12	55	41	53	64	61	61	66	84	111	113	150	186	192	166	147	180	188	167	140	124	119	90	87	90	94	107	114	77	86	92	98	80	74	77	60	48	38	32	6

EXHIBIT N

Data for the 57 City Problem

Source: Rand-McNally Road Atlas, 38th Edition, Rand-McNally Company: 1962

1	0																											
2	678	0																										
3	331	675	0																									
4	702	152	791	0																								
5	1210	1523	1531	1475	0																							
6	655	1074	399	1190	1818	0																						
7	210	881	364	900	1377	447	0																					
8	1334	1457	1655	1365	580	1942	1497	0																				
9	364	702	685	653	853	994	527	968	0																			
10	227	454	498	475	1187	869	425	1188	324	0																		
11	33	695	351	714	1187	631	186	1311	341	239	0																	
12	121	554	395	584	1205	763	327	1248	359	106	140	0																
13	1180	820	1422	665	1172	1821	1378	867	937	953	1254	1054	0															
14	1380	1411	1631	1302	681	1997	1507	101	1013	1172	1366	1236	784	0														
15	706	899	1027	823	672	1314	869	628	339	576	683	701	701	674	0													
16	190	709	511	730	1139	699	252	1263	293	255	167	185	1203	1318	635	0												
17	452	414	723	365	1114	1107	649	1076	292	225	463	331	749	1030	505	442	0											
18	202	605	556	591	1021	843	398	1099	175	151	212	154	997	1149	471	158	284	0										
19	301	729	80	816	1501	398	284	1625	655	461	321	382	1436	1619	997	481	713	526	0									
20	1806	2081	2127	2021	631	2414	1969	712	1449	1783	1783	1801	1577	811	1240	1735	1720	1617	2097	0								
21	1283	842	1457	666	1414	1856	1481	1109	1091	1056	1295	1162	242	1026	943	1311	843	1128	1482	1819	0							
22	291	510	568	487	1032	936	481	1079	184	109	295	173	881	1063	467	274	168	116	556	1629	1011	0						
23	913	402	1038	247	1453	1437	1102	1278	753	677	916	783	411	1195	818	924	495	763	1063	1972	429	647	0					
24	929	316	793	430	1839	1191	1083	1773	1027	776	960	859	1007	1727	1224	1031	739	927	865	2429	907	835	596	0				
25	656	680	934	586	928	1338	896	801	393	468	710	538	562	755	271	634	298	476	921	1457	785	365	563	996	0			
26	776	805	1053	713	789	1454	1009	652	504	596	823	658	495	606	206	775	424	587	1040	1308	737	485	664	1121	149	0		
27	237	731	558	721	1089	845	405	1209	239	270	219	232	1146	1261	578	84	414	130	528	1681	1252	246	893	1046	573	684	0	
28	847	521	1089	393	1193	1488	1045	1049	650	620	859	735	333	965	576	871	416	681	1075	1712	441	565	260	822	366	404	811	0
29	2371	2215	2649	2063	1659	3042	2597	1169	2092	2182	2411	2246	1405	1157	1794	2363	2012	2175	2628	1234	1547	2073	1816	2416	1735	1588	2272	1695
30	338	396	609	373	1146	980	536	1170	298	111	351	218	851	1124	572	367	125	218	572	1743	945	114	575	721	392	518	348	518
31	717	382	950	254	1248	1350	906	1111	538	481	720	596	472	1048	605	719	277	561	936	1767	566	445	213	679	350	459	691	139
32	452	793	773	739	771	1060	615	1003	91	429	429	447	1010	1002	356	381	378	263	743	1367	1164	271	830	1109	465	550	327	723
33	769	1080	1090	1037	431	1377	932	800	412	746	746	764	953	846	253	698	687	580	1060	1027	1195	592	1051	1396	509	458	644	829
34	513	257	728	205	1265	1162	718	1206	448	293	532	398	694	1160	629	541	160	400	738	1862	788	297	403	573	423	554	530	361
35	1043	518	1145	354	1623	1544	1241	1369	930	816	1055	928	503	1287	1006	1071	669	943	1170	2076	358	827	187	565	760	834	1060	430
36	487	862	187	978	1691	216	445	1811	841	653	507	551	1605	1866	1183	667	882	712	186	2273	1644	724	1225	980	1089	1209	714	1273
37	841	1014	1162	917	584	1449	1004	493	474	698	818	755	663	539	135	770	621	606	1132	1072	905	589	868	1325	353	204	713	608
38	451	656	766	617	858	1102	657	869	155	323	471	380	805	910	253	423	260	242	785	1456	973	214	661	972	243	352	369	530
39	406	772	97	888	1610	304	366	1730	760	572	426	470	1524	1785	1102	586	801	631	105	2202	1554	643	1135	890	1009	1128	633	1193
40	2011	1833	2289	1678	1500	2682	2237	920	1732	1824	2051	1886	1029	819	1441	2003	1652	1815	2268	1183	1156	1713	1440	2036	1375	1228	1912	1336
41	1127	1381	1448	1307	207	1735	1290	444	771	1104	1104	1122	965	522	492	1056	997	938	1418	739	1207	951	1258	1697	743	594	1002	998
42	105	714	230	751	1309	598	216	1429	459	284	125	182	1223	1484	801	285	509	330	200	1901	1326	356	956	897	720	840	332	899
43	2492	2682	2813	2572	1317	3100	2655	1207	2131	2395	2469	2455	2057	1285	1835	2421	2283	2303	2783	686	2235	2286	2468	2980	2008	1859	2367	2250
44	594	402	301	559	1794	701	616	1696	948	521	614	493	1224	1692	1090	774	671	638	376	2390	1225	629	806	492	960	1086	821	897
45	529	554	806	490	987	1239	797	906	294	340	611	411	643	860	335	512	170	354	794	1562	799	238	502	870	128	254	503	359
46	1792	1925	2113	1796	953	2400	1955	458	1425	1646	1769	1706	1253	507	1086	1721	1533	1557	2083	498	1428	1537	1664	2214	1260	1109	1667	1456
47	2554	2543	2875	2391	1646	3162	2717	1220	2187	2408	2531	2468	1763	1264	1848	2483	2295	2319	2845	1137	1946	2300	2174	2779	2023	1871	2429	2023
48	2416	2691	2737	2631	1223	3024	2579	1279	2055	2393	2393	2411	2129	1357	1850	2345	2330	2227	2707	610	2307	2239	2540	3007	2067	1918	2291	2322
49	2128	2403	2449	2343	935	2736	2291	1023	1767	2105	2105	2123	1885	1101	1562	2057	2042	1939	2419	322	2141	1951	2252	2719	1779	1630	2003	2034
50	475	603	760	563	921	1138	696	902	193	301	510	365	744	873	293	459	212	276	821	1507	900	192	588	919	200	311	373	457
51	750	666	1027	538	964	1460	1018	827	515	563	832	632	422	764	347	733	391	579	1015	1483	645	459	489	968	140	175	724	229
52	1106	467	984	547	1991	1379	1260	1903	1169	921	1137	1022	1090	1833	1366	1177	881	1072	1041	2555	980	978	679	200	1142	1244	1198	924
53	133	652	454	673	1089	741	296	1209	239	198	110	128	1149	1264	581	57	388	110	424	1681	1231	220	867	974	586	693	104	785
54	839	868	1120	773	751	1517	1072	597	567	659	886	721	517	538	268	838	487	654	1103	1262	738	552	727	1184	208	63	747	467
55	931	809	1208	683	972	1641	1199	770	696	744	1013	813	268	686	463	914	572	760	1196	1440	510	640	550	1112	321	257	905	290
56	331	637	38	753	1531	437	361	1655	685	492	351	395	1384	1710	1027	511	717	556	112	2127	1419	568	1000	755	934	1053	558	1051
57	971	922	1248	793	789	1649	1204	595	699	791	1018	853	383	508	401	970	619	782	1235	1256	625	698	720	1211	344	195	879	460
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28

EXHIBIT N
(Continued)

Source: Rand-McNally Road Atlas, 38th Edition, Rand-McNally Company: 1962

	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56
29	0																											
30	2106	0																										
31	1825	379	0																									
32	2138	387	618	0																								
33	1942	706	838	330	0																							
34	2047	182	222	534	847	0																						
35	1905	712	400	1017	1238	530	0																					
36	2797	764	1134	929	1246	915	1332	0																				
37	1669	694	663	491	359	751	1038	1318	0																			
38	1940	326	448	227	423	407	842	971	375	0																		
39	2716	682	1054	848	1165	825	1242	92	1237	890	0																	
40	390	1746	1467	1778	1657	1689	1514	2437	1313	1580	2356	0																
41	1556	1065	1053	700	393	1121	1415	1604	390	745	1528	1341	0															
42	2429	395	760	547	864	562	1092	386	936	589	305	2069	1222	0														
43	994	2377	2318	2053	1713	2413	2560	2959	1700	2076	2888	1273	1425	2587	0													
44	2583	568	758	1036	1353	559	919	488	1213	838	398	2225	1711	451	3076	0												
45	1842	264	289	367	549	300	689	962	453	174	885	1482	827	596	2113	832	0											
46	731	1627	1535	1460	1231	1663	1756	2269	951	1327	2188	685	845	1887	807	2154	1363	0										
47	405	2389	2153	2222	1993	2375	2264	3031	1713	2090	2950	791	1607	2649	669	2916	2125	762	0									
48	1170	2353	2377	1977	1637	2472	2632	2883	1682	2066	2812	1490	1349	2511	176	3000	2172	879	845	0								
49	1246	2065	2089	1689	1349	2184	2398	2595	1394	1778	2524	1368	1061	2223	364	2712	1884	721	922	288	0							
50	1899	279	375	265	486	359	769	1007	415	73	926	1539	785	548	2109	821	101	1360	2122	2117	1829	0						
51	1634	485	284	588	600	447	659	1183	379	383	1102	1276	769	903	2034	1003	221	1251	1961	2093	1805	322	0					
52	2489	864	785	1259	1547	724	646	1170	1448	1123	1080	2111	1838	1080	3103	682	1021	2340	2831	3165	2877	1070	1069	0				
53	2282	310	665	327	644	484	1014	610	716	369	529	1979	1002	228	2416	621	506	1667	2429	2291	2003	405	727	1120	0			
54	1532	585	522	612	521	620	902	1272	167	415	1191	1172	557	903	1801	1149	317	1049	1811	1872	1584	374	235	1307	756	0		
55	1448	666	429	769	715	628	689	1364	395	564	1283	1093	765	996	1970	1187	402	1171	1776	2042	1762	503	181	1214	908	228	0	
56	2649	598	912	773	1090	690	1107	225	1162	815	135	2289	1448	230	2813	263	806	2113	2875	2737	2449	851	1027	945	454	1116	1222	0
57	1393	713	532	745	653	703	834	1404	306	547	1323	1033	582	1036	1797	1251	449	1003	1721	1866	1578	506	248	1317	888	139	183	1248